A PROGRESSIVE DATA COMPRESSION SCHEME BASED UPON ADAPTIVE TRANSFORM CODING

Mixture Block Coding of Natural Images

Martin C. Ross and Khalid Sayood
Department of Electrical Engineering
University of Nebraska, Lincoln, NE 68588 0511

Abstract

A method for efficiently coding natural images using a vector-quantized variable-blocksize transform source coder is presented. The method, Mixture Block Coding (MBC), incorporates variable-rate coding by using a mixture of discrete cosine transform (DCT) source coders. Which coders are selected to code any given image region is made through a threshold driven distortion criterion. In this paper, MBC is used in two different applications. The base method is concerned with single-pass low-rate image data compression. The second is a natural extension of the base method which allows for low-rate progressive transmission (PT). Since the base method adapts easily to progressive coding, it offers the aesthetic advantages of progressive coding without incorporating excessive channel overload. Image compression rates of approximately 0.5 bit/pel are demonstrated for both monochrome and color images.

1 Introduction

Natural images contain regions of high and low detail. The regions of high detail are more difficult to code than those of low detail. Since the difference between number of bits required to code these two types of regions with acceptable distortion levels can be quite large, it is desirable to use a variable-rate source coding method. One way to do this is by using a transform coder which has more than one blocksize. Regions of high detail are coded with smaller blocks, while regions of low detail are coded with larger blocks. A smaller number of bits are used to code each of the different blocksize variable-rate coding is achieved. When trying to maximize the quality of the reconstructed image, better rates of coding bits can be achieved with the use of vector quantization. The goal is to describe a method for coding images using both vector quantization and multiphase recursive transform coding. A block-threshold technique is used to select the blocksize used to code any particular region of the image.

Both vector quantization and transform coding are block coding methods. Block coding methods are very useful when designing low-rate image compression systems, and are used almost exclusively when coding at low rates < 1 bit/pel. Traditional methods presented in the literature use either of these techniques, but have rarely used both together until very recently. One of the earlier publications to do this appeared in 1984 [1].

When using traditional vector quantizers for coding images, small blocksize are used to limit the size of the required quantizer codebook. But, when coding natural images with a very small number of bits it is desirable to use as large a blocksize as possible to take maximum advantage of the high inter-pixel correlations. In general, this blocksize is usually larger than vector quantization techniques can comfortably handle. One method to overcome this problem incorporates subsampled vector quantization [2], but little has been done to use traditional vector quantization with variable-rate coding.

Small codebooks are needed because the best performing vector quantization techniques (i.e., the LBG method [3]) are clustering techniques whose codebooks are very unstructured. As a result, the codebooks are difficult to construct and use. If a codebook is designed to be less computationally intensive, such as with lattice quantizers [4], or pyramid vector quantizers [5], the attainable distortion per codeword increases for a given coding rate.

Transform coding techniques easily allow for the use of large blocksize so high data compression ratios can be attained. But, it is difficult to keep good high-detail resolution when using large transform blocksize [6]. This is true since most methods code only the low-frequency high-energy transform coefficients [7]. As a result, the high-frequency coefficients are often ignored. Since these coefficients carry most of the information about the image's fine detail image quality suffers. Even when the image coefficients are vector quantized, so more coefficients can be coded for a given average rate, it can still be difficult to get high-detail resolution.

By using more than one blocksize, some of the inherent problems associated with low-rate transform coding can be overcome. Especially when vector quantizers are used to code the transform coefficients. The vector quantizer codebooks used here are of low dimensionality to keep their implementation simple. This is done by limiting the number of coefficients coded within any given block. For the examples given below, the vector quantizers code at most three coefficients as a vector for monochrome images, and nine coefficients (the transform pel) for color images. Never more than four transform pel's are coded within any block, no matter the size.

After a block is coded using these coefficient-limited transform coders the distortion is measured, to see if it meets a predetermined coding threshold. If a block coder poorly, it is divided into four smaller blocks and recoded until a distortion threshold is met or the maximum blocksize is attained. Thus, keeping the overall image integrity high by using the smaller blocks to more intensively code the high-detail regions.

In section 2 the threshold driven MBC coding algorithm is discussed. Also, the required overhead sent to the receiver to describe the final block structure of the coder is presented. Section 3 is a presentation of the MBC progressive transmission (MBC/PT) modification. In sections 4 and 5 the transform coder and the vector quantizers used in the example are shown. And finally, several examples are presented.

2 The Threshold Driven Structure of MBC

As mentioned above, each block of the image is coded using only a small number of transform coefficients. The difference between the original image and the coded image block is measured, and if the difference does not fall below a predetermined threshold, the block is divided into four smaller blocks and recoded. A new threshold is then applied to see if any of these blocks need to be divided further. This divide-and-test algorithm is continued until the entire image is coded with distortion that is less than the block threshold levels or the smallest blocksize is reached.

The monochrome images are coded using the maximum absolute difference distortion measure.

\[ d = \max_i |x_i - y_i| \]

where the range of \( x \) is taken over the image block being coded, and \( y_i \) is the coded value of pixel \( x_i \). For color images the maximum square difference is used,

\[ d = \max_i (x_i - y_i)^2 / (x_i - y_i)^2 / 3 \]

where \( y_i \) is the coded value of color pel \( x_i \).

\[ \text{This work was supported by the NASA Goddard Space Flight Center under grant NAG 5-916.} \]
To demonstrate the MBC method consider the coding of an example image segment. In the following it is assumed the image is coded with a starting blocksize of 16x16 and a smallest blocksize of 2x2.

First a 16x16 block is coded using the DCT method described below, and the distortion level for the block is measured. If this distortion is greater than the predetermined maximum level for 16x16 blocks, \( d_{\text{min}}(16x16) \), the block is divided into four 8x8 blocks for additional coding. After each of the 8x8 blocks is coded its resulting distortion levels are compared with the 8x8 distortion level, \( d_{\text{min}}(8x8) \). This process is continued until the only blocks not meeting their given distortion threshold are blocks of size 2x2. Since 2x2 is the smallest allowed blocksize, these blocks are transmitted de facto, making no further attempt to improve their distortion level.

Each 16x16 block can be completely coded, using all blocktrans, before moving onto the next 16x16 block or all the 16x16 blocks of the entire image can be coded before moving to the 8x8 blocks. For MBC, this sequence is inessential. The later method allows one to develop the progressive technique introduced in the next section.

Consider the example coding of a 16x16 image block shown in Figure 1. For clarity in the following material, let the four sub-blocks of an arbitrary block be numbered as shown in Figure 2 and let the block distortion thresholds be:

\[
d_{\text{min}}(16x16) = 12, \quad d_{\text{min}}(8x8) = 3 \quad \text{and} \quad \text{d}_{\text{min}}(2x2) = 10.
\]

After coding this example 16x16 block, say, to the 8x8 blocks, let the four coded 8x8 blocks have distortion levels, say, 3, 15, 4, and 6. Since each of these 8x8 blocks must be divided into four 4x4 blocks, it is necessary to divide and code as four 4x4. Notice the 4x4 distortion levels of this 8x8 block are 10, 8, 15, and 9, and it can be seen that only one of the 4x4 blocks fails to meet \( d_{\text{min}}(8x8) = 3 \). This block is coded as four 2x2 blocks.

The above coding process is depicted as a quad tree structure in Figure 3. Here there is

- one 16x16 block,
- four 8x8 blocks,
- four 4x4 blocks,
- four 2x2 blocks,

for a total of 16 blocks of coded information which are connected together through the use of a relative pointer scheme. To guarantee the receiver reconstructs the image correctly, a bit map of side information is sent along with the block coefficients information. One bit of side information is needed for each block size greater than 2x2. The bit map is divided into sub-blocks as the block is divided, with each sub-block's index set to 0. To tell the receiver the size of each coded block, 9 bits of side information are needed.

The first bit tells the 16x16 block is divided into 8x8 blocks. The next four bits indicate the 8x8 is divided into 4x4 blocks. The last four bits indicate that only the first 4x4 block is divided into 2x2 blocks. Notice the 2x2 blocks are grouped together, with the bit maps generated at the 4x4 level. This side information is coded.

If \( p_i \) is the number of pels and \( d_i \) is the number of bits used to code each block of that size, then the average coding rate for the blocks of that size is \( p_i d_i \text{ bit/pel} \). For the 2x2 block there are no overhead bits, so \( p_2 = 2, d_2 = 1 \). The average coding rate for the entire MBC system is

\[
\frac{\sum p_i d_i}{\sum p_i} = \frac{p_1 d_1 + p_2 d_2}{p_1 + p_2},
\]

(1a,b)

where \( p_1 \) is image information coded with \( p_2 \) pass blocksize.

There is no need to force the coding method of one blocksize onto another blocksize, as it is the case for the examples of this paper. Sometimes this may be desirable or, in fact, be impossible. Consider the case where blocks of size 16x16 are coded with, say, six DCT coefficients. The 8x8 and 4x4 blocks could be coded in a similar fashion but, of course, it is impossible to DCT code the 2x2 blocks with six transform coefficients. This indicates the 2x2 blocks must be coded with a different method. Likewise as indicated above, there is no reason to keep the distortion threshold levels constant. It is even possible to change the distortion measure for each blocksize. And since the different blocksize encompass different spatial frequency ranges it may be desirable to do this.

### Progressive Transmission MBC

Progressive transmission has grown out of need to transmit images over channels whose bandwidth is drastically smaller than what is available for the timely reconstruction of full-field imagery. Since slow-scan reception of images is nonaesthetic, the need to update the image on a frame-by-frame basis (or progressive basis) has arisen. In general, most methods found in the literature are concerned with perfect reconstruction of the image. The image is transmitted on a continually improving basis until it is completely transmitted so that it can be reconstructed without error. This requires the transmission of far more data than is needed to attain a visually pleasing reconstruction of the image (as is the case considered here). But, much of this literature is directly applicable to the low-rate transmission of images since almost every PT method reconstructs a visually acceptable field within a limited number of passes [e.g., 12].

In the previous section a single example 16x16 block was coded by passing through all of the necessary blocktrans before moving on to the next 16x16 block. But, if the entire image is passed through for each blocksize and the difference image is save for additional coding as is necessary in the next pass, the MBC method can be used as a PT coder. If each pass is immediately transmitted, the receiver can be reconstructing a crude representation of the image using these larger blocksize coefficients while the coder is processing the next pass. All passes after the first need only code the image residue. The residual coding information received in subsequent passes is added to the 'already waiting' image of the receiver. The image is updated using smaller blocks so it acquires more clarity with each pass. Since these blocks are of smaller size, each pass updates higher-frequency image components than were coded in the previous passes.

Since the first pass is coded with very few bits the receiver has an image, although a crude image, almost immediately. Since the successive passes are coding the difference image instead of the original there is no serious problem in updating the base MBC method for PT coding.

Since only image regions that code poorly are updated in successive passes, only those regions of the image which need additional coding will continue to use coder resources. This means that the regions of the image with low detail are coded quickly, and remain fixed, as the rest of the image continues to change as the information for each new pass is received.

The high detail regions of an image are coded more than once when using the MBC/PT method. Not only do the high detail regions require a greater channel capacity to transmit their coefficients because smaller blocktrans are being used but, it they also require channel resources in each of the previous passes. The rate for a MBC/PT coder is calculated using (1a), but the pass fractions are no longer constrained to add to one. In fact

\[
\sum p_i d_i, \quad \text{for} \ \text{each pass}.
\]

Thus, applying this to (1a) shows that it is possible to have \( p_1 d_1 > p_2 d_2 \). But this can be effected by the fact that MBC/PT may converge to the original image more quickly and require fewer blocks since the lower sections of the image coded with finer detail are taken from finer to a single pass. As is shown in section 5, an MBC/PT image can require fewer coding bits to transmit than image of similar quality using the MBC method.

### The Transform Coder

If a large block does not adequately code a given image region, it is divided into smaller blocks and recoded. So there is no strict advantage in using a large number of coefficients to code any particular blocksize. In fact, there is a tradeoff between expending more effort coding the larger blocks to fewer smaller blocks are used, and coding the larger blocks minimally so that the threshold algorithm assigns smaller blocks for coding.

For the examples of this paper it was chosen to code each block with only four DCT transform coefficients, including the dc and three lowest order frequency coefficients (Figure 4). This was done, not so much to attain the best overall coding rate, but to strike a median between PT coders which code a minimum of information about a given block [8] and those which code a large amount of information per block [9]. This accomplishes two things. Firstly, it shows an image can be adequately coded in a relatively small number of passes (four for the examples here) using a small number of transform coefficients. This each pass, and secondly, it shows that this can be done using a simple coding algorithm for each pass. In addition, when using the same transform coder for each pass it is also possible to use share the same vector quantizer between all of the passes. This saves quantizer design effort.
5 The Quantizers

The following four paragraphs discuss the quantizers used to code the examples discussed in section 5 of this paper. The remaining paragraphs discuss the details of LBG quantizers used.

When the MBC method was used to code monochrome images, the ac transform coefficients were quantized with an 8-bit linear scalar quantizer (LSQ), and the dc transform coefficients were quantized with a single 3-dimensional vector using an LBG vector quantizer whose codebook was of size 256.

When the MBC/PT method was used to code monochrome images, the quantizers for the first pass were different than those of subsequent passes. The these later passes code difference images that have nearly zero means blocks, while the first pass deals with the original image which is not zero mean. The dc coefficients for the first pass were coded with an 8-bit LSQ. In subsequent passes, the dc coefficients were coded with a 5-bit optimal laplacian scalar quantizer (OLSQ) [16]. The ac coefficients were coded the same for all passes using an LBG vector quantizer whose codebook contained 256 vectors.

The three non-dc coefficients of the YIQ images, when using the method MBC, were quantized with a vector quantizer whose codebook contained 1024 vectors. The de Y-components were quantized with an 8-bit LSQ and the de X and Q-components were coded with a 5-bit OLSQ.

As with the monochrome MBC/PT method, the YIQ MBC/PT images where coded differently in the first pass than in the subsequent passes. As was the case for MBC, the de Y-component was quantized with an 8-bit LSQ, while the de X and Q-components were quantized with a 5-bit OLSQ. In subsequent passes, the dc coefficients were quantized with an LBG vector quantizer whose codebook contained 64 vectors. The ac coefficients were quantized with the same LBG vector quantizer, whose codebook contained 1024 vectors, for all passes.

It was decided to use the same vector quantizer for the ac transform coefficients of each pass, no matter the blocksize. This rate was chosen because the ac coefficients for all passes were found to be distributed in a similar fashion. The only adjustment to be made was to allow for the differences in the coefficient variances, which had to be adjusted for each pass to guarantee a universal match the codebook variance.

By using this variance matching technique, more training vectors could be taken from a single image, so fewer images were needed to develop an adequate training set. The training set was built using scaling factors that mapped the unquantized coefficients of each pass into unit pdf. The scaling factors used are shown in Tables 1 and 2. The training vectors were then used to code the three non-dc coefficients of the YIQ images. They were built using a training set taken from three images different from the one coded for this paper.

The final vector quantizers were chosen to be non-adaptive. This is, the codebook was not modified to more effectively quantize output-bound source vectors at the coder moves from image to image [17]. This was done for two reasons. Sometimes the overhead required to implement such a technique can be overly expensive and the return acquired from it minimal. The extra effort needed stood against the design goal to construct a "simple to implement" vector-quantized adaptive transform source coder.

6 Results

All of the examples, as listed in Tables 3-6, use the 512×512 RGB woman/face picture of the UCLA database. The monochrome examples use the green (G) color field, while the YIQ images are made using the RGB to YIQ transformation matrix of [12]. All the examples use a starting blocksize of 16×16 and a final blocksize of 2×2. These tables list the number of blocks coded for each blocksize, and the thresholds used to test the quality of the coding passes. Also, the MBC/PT tables list the average coding rate that has accumulated after each pass.

This rate is based upon the average of the coding bits as spread across the entire image, without concern for what fraction of the image is coded within any particular pass. These rates represent the coding rate that is required to code the image if the coder where to stop with that particular pass. Since the remaining passes are yet to be coded, the image percentage coded within the indicated pass must be updated to include the image percentages coded in all of subsequent passes. For example, consider the MBC/PT rate of Table 4 when stopping at 4×4 blocks. In this case, the rate is .356 bits/pel, and the percentage of blocks coded with 4×4 blocks is 25.25 percent.

To code the monochrome image with MBC/PT only requires an extra .011 bits/pel over MBC. The overhead needed to code any given image is a function of the LBG codebook, and the codebook is a function of the training set used. A differently constructed codebook could offer different results. It is interesting to note that the YIQ MBC/PT method requires less coding rate to obtain the same image quality (PSNR) as is obtained when using MBC alone. It is clear that MBC/PT does not require excessive overhead to add the desirable PT feature.

References


Table 1. Vector quantizer scale factors for monochrome images.

<table>
<thead>
<tr>
<th>Blocksize</th>
<th>scale factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>16×16</td>
<td>60</td>
</tr>
<tr>
<td>8×8</td>
<td>35</td>
</tr>
<tr>
<td>4×4</td>
<td>20</td>
</tr>
<tr>
<td>2×2</td>
<td>10</td>
</tr>
</tbody>
</table>

Table 2. Vector quantizer scale factors for YIQ images.

<table>
<thead>
<tr>
<th>Blocksize</th>
<th>scale factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>16×16</td>
<td>60</td>
</tr>
<tr>
<td>8×8</td>
<td>35</td>
</tr>
<tr>
<td>4×4</td>
<td>20</td>
</tr>
<tr>
<td>2×2</td>
<td>10</td>
</tr>
</tbody>
</table>
Table 3. Monochrome MBC rate (bits/pel)

<table>
<thead>
<tr>
<th>min blocksize</th>
<th>#blocks</th>
<th>%image</th>
<th>PSNR</th>
<th>rate</th>
<th>threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x16</td>
<td>533</td>
<td>52.05</td>
<td>23.11 dB</td>
<td>33</td>
<td></td>
</tr>
<tr>
<td>8x8</td>
<td>1042</td>
<td>21.56</td>
<td>26.22</td>
<td>-</td>
<td>33</td>
</tr>
<tr>
<td>4x4</td>
<td>2510</td>
<td>16.01</td>
<td>22.61</td>
<td>-</td>
<td>33</td>
</tr>
<tr>
<td>2x2</td>
<td>2856</td>
<td>4.36</td>
<td>31.01</td>
<td>.473</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 4. Monochrome MBC/PT rate (bits/pel)

<table>
<thead>
<tr>
<th>min blocksize</th>
<th>#blocks</th>
<th>%image</th>
<th>PSNR</th>
<th>rate</th>
<th>threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x16</td>
<td>1001</td>
<td>100.00</td>
<td>23.13 dB</td>
<td>.063</td>
<td>33</td>
</tr>
<tr>
<td>8x8</td>
<td>1910</td>
<td>47.48</td>
<td>26.42</td>
<td>.170</td>
<td>31</td>
</tr>
<tr>
<td>4x4</td>
<td>3390</td>
<td>21.48</td>
<td>29.76</td>
<td>.368</td>
<td>33</td>
</tr>
<tr>
<td>2x2</td>
<td>2472</td>
<td>3.17</td>
<td>30.94</td>
<td>.400</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5. YIQ MBC rate (bits/pel)

<table>
<thead>
<tr>
<th>min blocksize</th>
<th>#blocks</th>
<th>%image</th>
<th>PSNR</th>
<th>rate</th>
<th>threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x16</td>
<td>569</td>
<td>57.62</td>
<td>21.99 dB</td>
<td>-</td>
<td>12</td>
</tr>
<tr>
<td>8x8</td>
<td>706</td>
<td>23.85</td>
<td>26.77</td>
<td>-</td>
<td>12</td>
</tr>
<tr>
<td>4x4</td>
<td>2414</td>
<td>19.09</td>
<td>29.04</td>
<td>-</td>
<td>11</td>
</tr>
<tr>
<td>2x2</td>
<td>2044</td>
<td>3.01</td>
<td>26.68</td>
<td>.691</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 6. YIQ MBC/PT rate (bits/pel)

<table>
<thead>
<tr>
<th>min blocksize</th>
<th>#blocks</th>
<th>%image</th>
<th>PSNR</th>
<th>rate</th>
<th>threshold</th>
</tr>
</thead>
<tbody>
<tr>
<td>16x16</td>
<td>3621</td>
<td>100.00</td>
<td>24.12 dB</td>
<td>.150</td>
<td>12</td>
</tr>
<tr>
<td>8x8</td>
<td>1790</td>
<td>52.97</td>
<td>26.42</td>
<td>.220</td>
<td>12</td>
</tr>
<tr>
<td>4x4</td>
<td>4560</td>
<td>17.31</td>
<td>22.09</td>
<td>.402</td>
<td>11</td>
</tr>
<tr>
<td>2x2</td>
<td>1413</td>
<td>2.72</td>
<td>26.63</td>
<td>.516</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 1. Example coded 16x16 block

Figure 2. Example Tree Structure

Figure 4. Lowest order DCT coefficients